# APPLICATION

# FOR

# UNITED STATES LETTERS PATENT

TITLE: COLLABORATIVE DESIGN PROCESS

APPLICANT: WOLFGANG KALTHOFF, GUENTER HUBER, GUIDO HOECKELE, THOMAS VOGT, AND BEATE KOCH

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EV321386557US

July 17, 2003

Date of Deposit

# COLLABORATIVE DESIGN PROCESS

## CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority based on United States Patent Application No. 10/306,718 for COLLABORATION PROCESS, filed November 27, 2002, which claims priority to Provisional United States Patent Application No. 60/367,397 for MASTER DATA MANAGEMENT, filed March 21, 2002, the disclosures of which are incorporated here by reference in their entirety.

## BACKGROUND

The present invention relates to data processing, and more particularly, to sharing data.

An important factor in the success of businesses in the current era is the ability to flexibly react to the requirements of the market. Shorter product cycles require decreased product development times and quicker introduction to market, while increased customer demand for individual solutions increases the number of variants.

These factors require businesses to be able to proceed simultaneously with the definition of the product requirements and its structure as well as the introduction of collaborative processes during communication with subcontractors and development partners. New products must be introduced rapidly to market based on internally and externally defined requirements. In order to do this, different approaches must be considered, and input can be provided by different groups of people. Thus, the data used in the collaborative processes has to be monitored for consistency.

For example, one process that requires collaboration is the integration of engineering partners into product development. Different groups of people can change the available data while working on their part of the project. This can lead to logical corruption of data, where two different copies of the same data are edited simultaneously, or changes made to related data no longer match. This logical corruption can render the data unreliable.

## SUMMARY

The present invention provides methods and apparatus, including computer program products, for sharing data in a product creation process.

In general, in one aspect, the invention features methods and apparatus for implementing a technique for sharing information. The technique includes defining a stored data set maintained by a first entity to include a locked data set and an unlocked data set and providing a second entity with access to the stored data set. The second entity has permission to view the locked data set and to change only the unlocked data set.

Particular implementations can include one or more of the following features. Providing a second entity with access to the stored data set can include providing an application in a computer system with access to the stored data set. The application can be maintained at a location external to the first entity. The application can include a computer aided design system. The locked data set can include information to call the application while the unlocked data set can include data to be used by the application. Alternatively, the locked data set can include version data for the application and the unlocked data set can include raw data for the second entity to look at or use.

The first entity can also be provided with access to the stored data set. The first entity can have permission to view the unlocked data set and to change only the locked data set. The second entity can include a computer aided design system.

Providing the second entity with access to the stored data set can include sending the data to the second entity. The second entity can include an entity that is external to the first entity. Data in the stored data set can be assigned to the locked data set and the unlocked data set based on predetermined criteria. The stored data set can also include a restricted data set including data that is not part of the locked data set or the unlocked data set. Data can be assigned to the locked data set based on closeness criteria. Closeness criteria can include geometric closeness, organizational closeness, and collective closeness. Data in the stored data set can be assigned to the locked data set and the unlocked data set based on a function of the second entity.

The stored data set can be defined to include a locked data set and an unlocked data set for the second entity where data included in the unlocked data set for the second entity can be defined as being locked for other entities.

The technique can also include transmitting data from the stored data set to the second entity, receiving modified data from the second entity, and integrating the modified data corresponding to the unlocked data set into the stored data set. The stored data set can be defined to include a locked data set and an unlocked data set based on user input.

In another aspect, the invention features methods and apparatus for implementing a technique for sharing information. The technique includes defining a master data set in a first entity, assigning permissions to a subset of data within the master data set based on predetermined criteria, transmitting a copy of the master data set with indications of the permissions to a second entity, and receiving manipulated master data set in accordance with the assigned permissions. The assigned permissions can include permission to change data that is a subset of the transmitted copy of the master data.

Particular implementations can include one or more of the following features. The technique also can include receiving a modified copy of the master data set from the second entity and integrating the modified copy of the master data set with the master data set. The modified copy of the master data set can include additional data and/or changed data. The changed data can include data that has been changed in response to testing.

The assigned permissions can include any combination of authority to read data, authority to change data, authority to add data and authority to delete data. The assigned permissions can also include authority to access predetermined types of data within the subset.

Assigning permissions can include assigning permissions based on at least one of an identity of an entity, a function of the entity and a user's position within the entity. Assigning permissions based on the user's position within the entity can include assigning permissions according to a hierarchy within the second entity so that a highest ranking member of an entity has a greater number of permissions, and a number and extent of permissions decrease

as rank decreases. Assigning permissions can also include assigning different permissions for different subsets of the unlocked data.

In yet another aspect, the invention features methods and apparatus implementing a technique for sharing information. The technique can include receiving, in a second entity from a first entity, a copy of a master data set with permissions for using the master data set, modifying the copy of the master data set according to the permissions, and transmitting the modified copy of the master data set to the first entity. The master data set can include locked and unlocked data.

Particular implementations can include one or more of the following features. Receiving the copy of the master data set in a second entity can include receiving the copy of the master data set in a computer application. Receiving the copy of the master data set in a computer application can include receiving version information regarding the computer application in the locked data and receiving raw data for manipulation in the unlocked data.

Modifying the copy of the master data set can include performing design processes on the unlocked portion of the data.

Receiving the copy of the master data set with permissions for using the master data set can include receiving permissions to do at least one of read, change and add data to the unlocked data. Receiving the copy of the master data set with permissions for using the master data set can also include receiving the copy of the master data set with permissions based on subsets of the unlocked data, with different permissions assigned for different subsets of the unlocked data.

Receiving the copy of the master data set with permissions for using the master data set can further include receiving the copy of the master data set with permissions based on at least one of an identity of the second entity, a function of the second entity and a hierarchy of users within the second entity.

The invention can be implemented to realize one or more of the following advantages. An entity can share data with other entities while ensuring that the data is consistent and reliable. By defining a master data set to include a locked data set and an unlocked data set, an entity that maintains a master data set can protect data in the master data set that it shares

with other users and/or applications. Locking data and maintaining a master data set also allows reduction of reconciliation processes and conserves the integrity of product data relationships. Since the master data tracks all changes, only an entity having the appropriate permissions can change the data. Thus, two versions of data can avoid having overlapping edits that need to be reconciled, and product relationships do not need to be reconciled.

Providing locked data with unlocked data, or data that an entity is permitted to edit, allows an entity to receive context information related to the data the entity is permitted to edit. Thus, an entity can see how modifications of the unlocked data will affect related data.

The details of one or more implementations of the invention are set forth in the accompanying drawings and the description below. Other features and advantages of the invention will become apparent from the description, the drawings, and the claims.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating a product creation system.

FIG. 2 is a block diagram illustrating a data sharing system.

FIG. 3 is a block diagram illustrating an alternative data sharing system.

FIG. 4 is a flow diagram illustrating a method of sharing data.

FIG. 5 is a flow diagram illustrating an alternative method of sharing data.

FIG. 6 is a flow diagram illustrating an implementation of a method of using shared data.

Like reference numbers and designations in the various drawings indicate like elements.

## DETAILED DESCRIPTION

As shown in FIG. 1, a data sharing system can be implemented in a product creation system 100. The product creation system 100 includes a central module 110 and entity modules 120. Each of the modules 110, 120 includes a data store 115, 125. Although the following description is directed to a product creation system, the data sharing system 100 can be used with any system in which data is shared by two entities.

The central module 110 represents a centralized control for a product creation process. The entity modules 120 represent entities involved in the product creation process. The entities can include computer applications as well as companies, departments or individuals. Many different entities can participate in the development of a product. The entities can include internal and external entities. For example, entity modules 120 can include modules for design, testing, purchasing, marketing, sales, manufacturing, installation, customer service, technical services, customers, dealers, distributors, suppliers, vendors, and regulatory organizations. The central module 110 and the entity modules 120 can communicate through a network 180.

The central module 110 includes a central data store 115 that stores master data for a product creation process. The master data includes objects associated with the product creation process. These objects can include, for example, product classes, attributes, product line designs, parts lists, materials lists, quality specifications, requirements, routing and process structures.

The central data can be accessed by the entity modules 120, based on permissions, or transmitted to the entity modules 120 through network 180. Each of the entity modules 120 includes a data store 125 for storing the data received from the central module 110 and for additional entity specific data developed in entity module 120. For example, if the entity module 120 is a marketing module, the corresponding data store 125 could store marketing specific information.

The entity modules 120 can develop objects of the product creation process in parallel. Each entity module 120 can develop different versions of each object of the product creation process. For example, classes of a product can include types, models, body styles, sizes, etc. Attributes can include information about component parts such as engine type (e.g., 95 HP, 110 HP, 125 HP, etc.) or brake type (e.g., disc or drum), or characteristics such as body color (e.g., black, silver, red, etc.) or fuel efficiency (e.g., 30 m.p.g.).

FIG. 2 is a block diagram illustrating a data sharing system 200. The data sharing system 200 includes a set of master data 212 stored in a first entity data store, such as central data store 115. The first entity can include central module 110. The master data 212 can be

shared with a second entity 120 by allowing access to the second entity, such as an entity module 120. The data sharing system 200 allows master data 212 to include a locked data set 250 and an unlocked data set 260. Data within the master data 212 can be assigned to the locked data set 250 and unlocked data 260 by the first entity or users associated with the first entity.

The master data 212 can also include a set of restricted data 240. The restricted data 240 can be assigned by the first entity. The restricted data includes all data that is not included in the locked data set 250 or the unlocked data set 260. The restricted data set 240 can include data that is not relevant to the second entity 120 or confidential data to which the second entity 120 is not given access.

Unlocked data set 260 can include data that is to be manipulated by the second entity. When a specified entity is assigned a specified unlocked data set 260, the specified unlocked data set 260 in the first entity can be configured so that other entities will not be able to change the data in the specified unlocked data set 260. For example, the data in the specified unlocked data set 260 assigned to the specified entity can be designated as locked in the first entity, and can be assigned to other entities as locked data 250. If the specified entity updates data in the specified unlocked data set 260, the corresponding data that was assigned to other entities as locked data 250 will also be updated. The update to locked data 250 in other entities can be performed immediately through a standing link.

The locked data set 250 can include information that the second entity has access to but does not have permission to change. For example, if the second entity can be a computer aided design ("CAD") system. A CAD system can be used by a designer to view graphic representations of a design, based on design data, from various angles and at various zoom distances. Thus, if the second entity is a CAD system, the unlocked data set 260 can include design related data while the locked data set 250 can include information that the first entity uses to call the testing CAD system. The CAD system can access the unlocked data set 260 and manipulate the data, such as by creating graphic representations of a design using the data and updating the data in the unlocked data set 260, for example, if a user makes changes in the design.

The unlocked data set 260 can include a variety of permissions for manipulation. The permissions can include permissions to read the data, add new data within the unlocked data set 260 (e.g., add a new category of data), and permission to change or delete existing data in the unlocked data set. The unlocked data set can be divided into smaller subsets where the permissions vary from set to set. For example, the entity can read one subset, change data in another subset, and add and change data in a third subset.

The permissions can be assigned to the second entity according to a hierarchy. For example, if the second entity is an entity including a plurality of employees, the permissions can be assigned according to position within the entity so that a highest ranking member of the entity has a greater number of permissions, and a number and extent of permissions decrease as rank decrease. For example, the director of a group could have permission to read, add and change data in all subsets of the unlocked data set, while supervisors for departments within the group can have permission to read, add and change subsets of data that apply to their departments only. Lower level employees can be given permission to only read data within the subset of the unlocked data that applies to their departments. In addition to permissions, locked data could include identifiers for versions of applications or even applications to use in manipulating the data (the unlocked data).

The permissions can also be used to divide the master data set 212 into the locked data set 250, unlocked data set 260 and the restricted data set 240. For example, data that is made available to the second entity 120 with an assigned a permission of read only can be part of the locked data set 250. Data that is assigned a permission allowing the second entity 120 to change the data can be part of the unlocked data set 260.

Data can be separated between the locked data set 250 and the restricted data set 240 based on closeness criteria. Closeness criteria can include geometrical closeness, organizational closeness, and/or collective closeness. An algorithm for including data in the locked data set 250 that is based on geometrical closeness can cause the locked data set 250 to include data about items that are geometrically (e.g., physically) close to the subject of the unlocked data set 260. For example, if the unlocked data set 260 includes data regarding design of headlights of a car, the locked data set 250 can include information regarding the

headlight support on the body of the car. Because the headlight support is physically close to the headlight, and the headlight support can affect headlight design. Information regarding the car's rear wheel can be left in the restricted data set 240 because the rear wheel is not close to the headlight.

An algorithm based on organizational closeness can cause the locked data set to include data regarding items that are a part of the same organizational structure as the subject of the unlocked data set 260. For example, if the unlocked data set 260 includes data about the motor of a car, the locked data set 250 can include information regarding the rear axle of the car. Although the motor and the rear axle may not be geometrically close, they are part of the same organizational structure.

An algorithm based on collective closeness can cause the locked data set 250 to include data regarding items that are a part of the same data collection as the subject of the unlocked data set 260. Thus, if the unlocked data set 260 includes data regarding a car's headlights, the locked data set can include information regarding the car's battery. Although the battery is not geometrically close to the headlights, or part of the same organizational structure, the headlight and battery can be part of the same data collection. Thus, collective closeness can be used to override exclusions based on geometric or organizational closeness criteria.

Inclusion of data in the locked data set 250 can be limited by permissions, as discussed above. If data that can be included in the locked data set 250 based on the closeness criteria is considered too confidential to be shared with a second entity 120, the confidential data is excluded from the locked data set 250 and remains in the restricted data set 240.

FIG. 3 is a block diagram illustrating an alternative data sharing system 300. The first entity data store 315 includes locked data 350a and unlocked data 360a. The first entity 310 can use or change unlocked data 360a, but can only view the locked data 350a. The first entity 310 can transmit a copy of the master data 212 to the second entity 320.

The second entity 320 receives a copy of the master data 212 from the first entity 310 over network 380. The second entity 320 stores the received copy in a data store 325. When

the copy of the master data set 312 is transmitted to the second entity 320, the locked and unlocked data become reversed in the second entity data store 325. The locked data set 350a in the first entity 310 becomes the unlocked data set 360b in the second entity 320, while the unlocked data set 360a in the first entity 310 becomes the locked data set 350b in the second entity 320.

The second entity 320 can manipulate and modify the copy of the master data 212 in its data store 325 in accordance with permissions attached to the data. The second entity 320 transmits the modified copy of the master data 312 to the first entity 310. The first entity 310 can integrate the modified copy of the master data 312 with the copy of the master data 312 stored in data store 315 according to the permissions assigned. For example, if the modified copy of the master data 312 includes changes or new data from sources that did not have permission to manipulate the data, those modifications are not included in the master data 312 stored in data store 315.

FIG. 4 is a flow diagram illustrating a method of sharing data. The method includes defining a locked data set 250 and an unlocked data set 260 in a master data set 212.. (step 410) Defining the locked data set 250 and unlocked data set 260 can include defining the locked data set 250 and unlocked data set 260 based on permissions, as described above. Thus, data that is in the locked data set 250 for one entity 120 can be in the unlocked data set 260 for another entity 120.

The locked set of data 250 and the unlocked set of data 260 can also be divided based on the function of the entity 120 receiving the data. For example, if the entity 120 receiving the data is the marketing department, the unlocked set of data 260 can include marketing related information, such as marketing statistics or sales projections.

When the locked set of data 250 and the unlocked set of data 260 have been defined, a first entity 110 can provide a second entity 120 with access to the master data set 212 based on the permissions assigned to the second entity 120. (step 420) The second entity 120 can include, for example, an organization, an individual, a computer application, or any other entity capable of using data. The second entity 120 can manipulate the unlocked data set 260 according to the permissions assigned to it.

If the second entity 120 includes a computer application, the locked data set 250 can include information to call the application while the unlocked data set 260 includes data to be used by the application. For example, the locked data set 250 can include version data of the application called while the unlocked data set 260 includes raw data for the application to look at or use in its execution.

FIG. 5 is a flow diagram of another implementation of a method of sharing master data 212. The method can include transmitting the set of master data 212 from a first entity 110, such as central module 110, to a second entity 120, such as entity module 120. (Step 510) The master data 212 can be defined as described above with reference to FIG. 4.

The method further includes receiving a modified copy of the master data from the second entity 120. (Step 520) The modified copy of the master data can include additions or changes to the master data 212 performed by the second entity 120 according to permissions assigned to the second entity 120. When the modified copy of the master data is returned from the second entity 120 to the first entity 110, the locked and unlocked designations are returned to the first entity 110. Thus, data that was locked in the first entity 110, because it was available to the second entity 120 as unlocked data, can now be designated as unlocked in the first entity 110.

The first entity 110 integrates the modified copy of the master data into the master data set 212 stored in central data store 115, according to the permissions assigned to the second entity 120. (Step 530) For example, the first entity 110 can replace data in the unlocked data set 260 with modifications made in the unlocked data set 260 by the second entity 120. The first entity 110 can further check the source of each modification to determine if the source of the change in the second entity 120 had permission to modify the data. Then, the first entity 110 can modify the data in the unlocked data set 260 in central data store 115 only if the source had permission.

FIG. 6 is a flow diagram illustrating an implementation of a method of using shared master data 212 by the second entity 120. The method of using the shared master data 212 can include receiving a copy of the master data set 212 with permissions. (Step 610) The shared master data 212 is received in the second entity 120 from the first entity 110.

The second entity 120 can modify the copy of the master data set 212 according to the permissions received with the copy of the master data set 212. (Step 620) For example, if the second entity 120 is a CAD system, the second entity 120 can create graphic representations for a designer to view using the unlocked data set 260, and allow update of the design data.

After the second entity 120 is finished manipulating the copy of the master data set 212, the second entity 120 transmits the modified copy of the master data set to the first entity 110 (Step 630). The first entity 110 integrates the modified copy with the master data set in central data store 115, as described with reference to FIG. 5.

The invention can be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. The invention can be implemented as a computer program product, i.e., a computer program tangibly embodied in an information carrier, e.g., in a machine-readable storage device or in a propagated signal, for execution by, or to control the operation of, data processing apparatus, e.g., a programmable processor, a computer, or multiple computers. A computer program can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program can be deployed to be executed on one computer or on multiple computers at one site or distributed across multiple sites and interconnected by a communication network.

Method steps of the invention can be performed by one or more programmable processors executing a computer program to perform functions of the invention by operating on input data and generating output. Method steps can also be performed by, and apparatus of the invention can be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit).

Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data

from a read-only memory or a random access memory or both. The essential elements of a computer are a processor for executing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto-optical disks, or optical disks. Information carriers suitable for embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in special purpose logic circuitry.

To provide for interaction with a user, the invention can be implemented on a computer having a display device such as a CRT (cathode ray tube) or LCD (liquid crystal display) monitor for displaying information to the user and a keyboard and a pointing device such as a mouse or a trackball by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, such as visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input.

The invention can be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or an Web browser through which a user can interact with an implementation of the invention, or any combination of such back-end, middleware, or front-end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network ("LAN"), a wide area network ("WAN"), and the Internet.

The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

The invention has been described in terms of particular embodiments. Other embodiments are within the scope of the following claims. For example, the steps of the invention can be performed in a different order and still achieve desirable results. What is claimed is: